

APPLICATION FOR UNITED STATES LETTERS PATENT

FOR

METHOD AND SYSTEM TO ORDER MEMORY OPERATIONS

Inventor(s): Dennis M. O'Connor
Michael W. Morrow

Prepared by: Anthony M. Martinez,
Patent Attorney

intel®

Intel Corporation
5000 W. Chandler Blvd., CH6-404
Chandler, AZ 85226-3699
Phone: (480) 552-0624
Facsimile: (480) 554-7738

Docket #: P18395

"Express Mail" Label Number EL 962029012 US

METHOD AND SYSTEM TO ORDER MEMORY OPERATIONS

BACKGROUND

Various memory ordering schemes may be implemented in a computing system to address when a processor in a multiprocessor system "sees" memory operations by other processors. Memory ordering may also be referred to as memory consistency or event ordering. Memory operations, such as a load operation or store operation, may be seen at different times by different processors. This may lead to software not executing as expected or operating differently on a multiprocessor system compared to a uniprocessor system.

To address memory consistency, some memory consistency models have been developed. The different models have tradeoffs in terms of system performance.

Thus, there is a continuing need for alternate ways to implement memory consistency in a system.

BRIEF DESCRIPTION OF THE DRAWINGS

The subject matter regarded as the invention is particularly pointed out and distinctly claimed in the concluding portion of the specification. The present invention, however, both as to organization and method of operation, together with objects, features, and advantages thereof, may best be understood by reference to the following detailed description when read with the accompanying drawings in which:

FIG. 1 is a block diagram illustrating a computing system in accordance with an embodiment of the present invention; and

FIG. 2 is a block diagram illustrating a wireless device in accordance with an embodiment of the present invention.

It will be appreciated that for simplicity and clarity of illustration, elements illustrated in the figures have not necessarily been drawn to scale. For example, the dimensions of some of the elements are exaggerated relative to other elements for clarity. Further, where considered appropriate, reference numerals have been repeated among the figures to indicate corresponding or analogous elements.

DETAILED DESCRIPTION

In the following detailed description, numerous specific details are set forth in order to provide a thorough understanding of the present invention. However, it will be understood by those skilled in the art that the present invention may be practiced without these specific details. In other instances, well-known methods, procedures, components and circuits have not been described in detail so as not to obscure the present invention.

In the following description and claims, the terms "include" and "comprise," along with their derivatives, may be used, and are intended to be treated as synonyms for each other. In addition, in the following description and claims, the terms "coupled" and "connected," along with their derivatives, may be used. It should be understood that these terms are not intended as synonyms for each other. Rather, in particular embodiments, "connected" may be used to indicate that two or more elements are in direct physical or electrical contact with each other. "Coupled" may mean that two or more elements are in direct physical or electrical contact. However, "coupled" may also

mean that two or more elements are not in direct contact with each other, but yet still co-operate or interact with each other.

FIG. 1 is a block diagram illustrating a computing system 100 in accordance with an embodiment of the present invention. System 100 may include processors 110, 120, and 130. System 100 may further include a local cache memory 140 coupled to processor 110, a local cache memory 150 coupled to processor 120, and a local cache memory 160 coupled to processor 130. In addition, computing system 100 may further include a shared cache memory 170, wherein shared cache memory 170 is coupled to processor 110 via local cache 140, coupled to processor 120 via local cache 150, and coupled to processor 130 via local cache 160. Shared cache memory 170 may be coupled to local cache memories 140, 150, and 160 via a bus 180.

Processors 110, 120, and 130 may include logic to execute software instructions and may also be referred to as cores, controllers or processing units. Although system 100 is shown as having three processors, this is not a limitation of the present invention. In other embodiments, system 100 may include more or fewer processors.

Cache memories 140, 150, and 160 may be level 1 (L1) cache memories and cache memory 170 may be a level 2 (L2) cache memory. Cache memories 140, 150, 160, and 170 may be volatile or nonvolatile memories capable of storing software instructions and/or data. Although the scope of the present invention is not limited in this respect, in one embodiment, cache memories 140, 150, 160, and 170 may be volatile memories such as, for example, a static random access memory (SRAM) or a dynamic random access memory (DRAM).

In one embodiment, processors 110, 120, 130 may be integrated together as

part of a chip-level multiprocessor (CMP) system that has multiple processors or cores on a single silicon die. In an alternate embodiment, processors 110, 120, and 130 may be discrete components located on a motherboard. Cache memories 140, 150, 160, and 170 may be integrated with processors 110, 120, and 130 or may be external ("off-chip") components.

Cache memories 140, 150, 160, and 170 may collectively serve as the memory space or address space of system 100. Processors 110, 120, and 130 may use an address to access information from the memory space. In one embodiment, a 32-bit address may be used to access, e.g., read or write, information from a particular location in the memory space.

In one embodiment, system 100 may include six signal lines, 191, 192, 193, 194, 195, and 196. These signal lines may be communication paths to communicate information between the components of system 100. The signal lines may also be referred to as data lines or data paths, and may be coupled together in a wired-OR configuration.

As is shown in FIG. 1, signal line 191 may be labeled as REGION 0 LOAD_IF, signal line 192 may be labeled as REGION 0 STORE_IF, signal line 193 may be labeled as REGION 0 SWAP_IF, signal line 194 may be labeled as REGION 1 LOAD_IF, signal line 195 may be labeled as REGION 1 STORE_IF, and signal line 196 may be labeled as REGION 1 SWAP_IF. As is discussed below, signal lines 191-196 may be used to communicate the status of a memory operation to a particular region of memory. For example, signal line 191 may be asserted to indicate that a particular kind of memory operation, e.g., a load issued by processor 110 to region 0 of memory, is not

globally observable by all processors of system 100 but is observable by processor 110.

In one embodiment, signal lines 191-196 may be used for memory ordering in system 100. Memory ordering may refer to the order which memory operations are "seen" to happen, e.g., when a processor in a multiprocessor system such as system 5 100 "sees" memory operations by other processors of system 100. For example, a store operation may be "seen" when a load operation to the same location returns the value stored by that store, or some later store. A load operation may be "seen" when no subsequent store can effect the value returned by that load. Memory operations may be observed by components other than processors of system 100, e.g., a 10 peripheral such as a graphics controller (not shown) may also see a memory operation.

In multiprocessor systems, load and store operations might be each "seen" at different times by different processors. This can lead to software not executing the same way, or operating erratically, on a multiprocessor system compared to a uniprocessor system, depending on what guarantees about when operations are seen 15 by other processors are made by the hardware. The more guarantees the hardware makes, the easier it may be for programmers to create software that works as intended. Memory ordering models that offer more guarantees may be referred to as "strong" models and models that provide few or no guarantees may be referred to as "weak" models.

20 In an embodiment, a signaling mechanism may be used in system 100 to provide the guarantees and to provide a relatively strong memory ordering model. The signaling mechanism may be referred to as a sideband signaling mechanism and the signals may be referred to as sideband signals.

To implement the signaling mechanism, system 100 may include signal lines 191-196 to communicate the status of a memory operation to a particular region of memory. In addition, processors 110, 120, and 130 may include logic circuitry to generate, receive, and process the memory operation status signals. These signals
5 may be transferred between the components of system 100 using signal lines 191-196. Processors 110, 120, and 130 may assert signal lines 191-196 to communicate the status of a memory operation. In other words, processors 110, 120, and 130 may generate or assert signals to communicate the status of a memory operation.

In an embodiment, the memory or memory space of system 100 may be divided
10 into one or more regions and the ordering guarantees may then be enforced on a region-by-region basis. The memory may be divided into regions by using one or more bits from the memory address of the memory operation to determine the region of memory and these bits may be from anywhere in the address. In various embodiments, the memory of system 100 may be divided into 4, 8, 16, or 32 regions,
15 although the scope of the present invention is not limited in this respect.

In an embodiment, there may be three global signals for each region of memory, although the scope of the present invention is not limited in this respect. These three global signals may be referred to as a load-in-flight signal, a store-in-flight signal, and a swap-in-flight signal. These signals may indicate that a memory operation of the
20 specified type (e.g., load, store, or swap) has been "seen" by one processor but not seen by all processors, a condition that may be referred to as "in-flight" or "in progress." The term "globally observable" may be used to refer to when a memory operation is seen or may be seen by all processors of system 100. Accordingly, a memory

operation may be in-flight after it is issued by a processor and up until the point when it is seen or may be seen by all processors of system 100, i.e., up until the point when it is globally observable.

5 A swap operation may refer to a memory operation that includes both a store and a load and may also be referred to as an atomic swap. In a chip-level multiprocessor (CMP) system, the processors may be able to communicate “in-flight” status changes relatively quickly, e.g., within one processor clock cycle.

10 In one embodiment, if a load operation is in-flight to a particular region of memory, no component of system 100 will be allowed to issue any other memory operation other than a load operation into that region. For example, if a load operation is in-flight to a particular region of memory, all processors of system 100 will be prevented from issuing a store or swap into that region. This may guarantee that no other memory operation such as, for example, a store or swap, will alter the result of the load operation in the time between when it is “seen” by the first processor to handle it and the time when it is handled by the last processor to handle it, which results in all
15 processors effectively “seeing” the load at the same time. In this embodiment, while a load is in-flight in a particular region, other loads to that region may be allowed to issue since the contents of particular region of the memory of system 100 may not be altered by, e.g., any other store or swap operation, which will be prevented from issuing while a
20 load operation is in-flight.

Similarly, if a store operation is in-flight to a particular region memory, no component of system 100 will be allowed to issue any other memory operation other than a store operation into that region. For example, if a store operation is in-flight to a

particular region of memory, all processors of system 100 will be prevented from issuing a load or swap into that region. This may guarantee that all processors will “see” the store effectively at the same time. In this embodiment, while a store is in-flight to a particular region, other stores to that region may be allowed to issue to that region.

- 5 Since no other component in system 100 is allowed to read, e.g., issue a store or swap, a region a memory while a store is in-flight to that region, other stores to that region may be allowed while the store is in-flight.

In addition, if a swap operation is in-flight to a particular region of memory, no component of system 100 will be allowed to issue any memory operation to that region.

- 10 For example, if a swap operation is in-flight to a particular region of memory, all processors of system 100 will be prevented from issuing a load, store, or swap to that region. This may guarantee that all processors will “see” the swap effectively at the same time.

- Memory operations in one region may have no effect on memory operations in
15 other regions. By operating in this manner, system 100 may provide a relatively strong memory ordering model.

- No arbitration or handshaking between the processors may be necessary if the processors of system 100 communicate memory in-flight status to other processors in less than one instruction issue period, i.e., the amount of time it takes between one
20 instruction issuing and the next possible instruction issuing (e.g., one clock cycle). For example, in a chip-level multiprocessor system where multiple processors are integrated together on a single silicon die, a processor of the system may be able to assert a memory in-flight signal and have this signal recognized by other processors in

time to prevent the other processors from issuing a memory command. Since the in-flight signals may be logically OR signals or arranged in a wired-OR configuration, multiple processors may all assert an in-flight line simultaneously. So, even if one processor issues a store and another processor issues a load, this may be allowed
5 since while those load and store operations are in-flight, the processors will prevent any other subsequent stores or loads to issue since the in-flight signal lines are asserted.

In one embodiment, processors 110, 120, and 130 may include logic to generate or assert in-flight status signals and communicate this information to other components of system 100 using signal lines 191-196. In other words, processors 110, 120, and
10 130 may include logic to assert a particular in-flight status signal line depending on the memory operation issued by the processor and the region to which the memory operation is directed. In addition, processors 110, 120, and 130 may include logic to determine whether or not a particular signal line is asserted or deasserted, and logic to prevent itself from issuing a memory operation if the signal line is asserted.

15 Although the scope of the present invention is not limited in this respect, in an embodiment, the portion of system 100 in which an in-flight operation is currently being processed may be required to maintain the proper memory operation in-flight status for a particular region of memory. In an alternate embodiment, the device that issued the memory operation may have responsibility for asserting or deasserting the appropriate
20 signal line. And after the memory operation is satisfied, the portion of system 100 that satisfied the request may notify the device that the memory operation is satisfied and at this point the device that issued the operation can deassert the line.

In an embodiment, one processor of system 100 may completely handle a

memory operation, e.g., a load or a store, from the point when it is issued to the point when it is globally observable. In this embodiment, after the processor issues the memory operation, the processor may assert the appropriate signal line to indicate that the memory operation is in-flight, and then may deassert the signal line after the

5 memory operation is globally observable.

In an embodiment, the memory operation may be processed by more than one part of system 100. In this embodiment, responsibility for asserting the signal lines may be transferred to the part of system 100 processing the memory operation. For example, after a processor issues a memory operation, the next level of the cache

10 hierarchy may receive the memory operation and then responsibility for asserting the appropriate signal to indicate that the memory operation is in-flight may be transferred to this level of cache hierarchy.

In system 100 illustrated in FIG. 1, the processors may be responsible for maintaining the memory in-flight signaling for two regions of memory. After the memory

15 hierarchy has taken a memory operation to a point where the memory operation is completed, i.e., no longer in-flight, it has to signal the processors, which may then deassert the signal line or the processors will not hold the signal line asserted for that operation but may continue to assert the signal line for other memory operations.

Although system 100 is illustrated as including six status signals lines, this is not

20 a limitation of the present invention. More or fewer signals lines may be used. For example, in one embodiment, only one signal line may be used to indicate that a particular kind of memory operation, e.g., a load, store, or swap operation, is not globally observable in system 100 but is observable by at least one processor of system

100. This signal line may be asserted by the processor after the processor issues the memory operation.

In an alternate embodiment, the in-flight status may be tracked for each address of the memory space. For example, if a load is in-flight to a particular address, system 100 may prevent other types of memory operations other than load operations from issuing to this address.

In another embodiment, the memory space of system 100 may be divided into four regions. In this embodiment, 12 wires may be used to communicate whether a load, store, or swap is in-flight in each of the four regions of memory. Prior to issuing a memory operation, a processor may look at the address bits to decide what region the memory operation is targeted for and then may check the three in-flight lines for that region to see if any memory operation is in-flight to that region that would prohibit the processor from issuing a memory operation to that region. For example, bits 5 and 6 of a 32-bit memory address may be checked by the processor prior to issuing a memory operation.

As discussed above, the number of regions of memory is not a limitation of the present invention. In-flight status may be communicated for a single region of memory, or the memory may be divided into more than one region. For example, the memory space may be divided into two regions, wherein the odd addresses may form one region of memory and the even addresses may form another region of memory and three in-flight signal lines may be implemented for each region of memory. In another embodiment, the memory may be divided by odd sets and even sets in a set associative cache.

Accordingly, as is discussed above, an embodiment may include a method and apparatus to notify components of system 100 when and where memory operations are in-flight . This may speed up how quickly memory operations are seen or observed in system 100.

5 As is discussed, in one embodiment, a system to order memory operations is provided, wherein the system includes a processor to use at least one signal for memory consistency, wherein the at least one signal indicates that a particular kind of memory operation is not globally observable in the system but is observable by at least one processor of the system.

10 Turning to FIG. 2, shown is a block diagram of a wireless device 200 with which embodiments of the invention may be used. In one embodiment, wireless device 200 may include computing system 100 that is discussed above with reference to FIG. 1.

 As is shown in FIG. 2, wireless device 200 may include an antenna 210. In various embodiments, antenna 210 may be a dipole antenna, helical antenna or
15 another antenna adapted to wirelessly communicate information.

 Wireless device 200 may be a personal digital assistant (PDA), a laptop or portable computer with wireless capability, a web tablet, a wireless telephone (e.g., cordless or cellular phone), a pager, an instant messaging device, a digital music player, a digital camera, or other devices that may be adapted to transmit and/or
20 receive information wirelessly. Wireless device 200 may be used in any of the following systems: a wireless personal area network (WPAN) system, a wireless local area network (WLAN) system, a wireless metropolitan area network (WMAN) system, or a wireless wide area network (WWAN) system such as, for example, a cellular

system.

An example of a WLAN system includes a system substantially based on an Industrial Electrical and Electronics Engineers (IEEE) 802.11 standard. An example of a WMAN system includes a system substantially based on an Industrial Electrical and Electronics Engineers (IEEE) 802.16 standard. An example of a WPAN system includes a system substantially based on the Bluetooth™ standard (Bluetooth is a registered trademark of the Bluetooth Special Interest Group). Another example of a WPAN system includes a system substantially based on an Industrial Electrical and Electronics Engineers (IEEE) 802.15 standard such as, for example, the IEEE 802.15.3a specification using ultrawideband (UWB) technology.

Examples of cellular systems include: Code Division Multiple Access (CDMA) cellular radiotelephone communication systems, Global System for Mobile Communications (GSM) cellular radiotelephone systems, Enhanced data for GSM Evolution (EDGE) systems, North American Digital Cellular (NADC) cellular radiotelephone systems, Time Division Multiple Access (TDMA) systems, Extended-TDMA (E-TDMA) cellular radiotelephone systems, GPRS, third generation (3G) systems like Wide-band CDMA (WCDMA), CDMA-2000, Universal Mobile Telecommunications System (UMTS), or the like.

Although computing system 100 is illustrated as being used in a wireless device in one embodiment, this is not a limitation of the present invention. In alternate embodiments system 100 may be used in non-wireless devices such as, for example, a server, a desktop, or an embedded device not adapted to wirelessly communicate information.

While certain features of the invention have been illustrated and described herein, many modifications, substitutions, changes, and equivalents will now occur to those skilled in the art. It is, therefore, to be understood that the appended claims are intended to cover all such modifications and changes as fall within the true spirit of the

5 invention.